Guidelines for Designing IT Security Management Tools

Pooya Jaferian, David Botta, Fahimeh Raja, Kirstie Hawkey, Konstantin Beznosov
University of British Columbia, Vancouver, Canada
{pooya, botta, fahimehr, hawkey, beznosov}@ece.ubc.ca

ABSTRACT

An important factor that impacts the effectiveness of security systems within an organization is the usability of security management tools. In this paper, we present a survey of design guidelines for such tools. We gathered guidelines and recommendations related to IT security management tools from the literature as well as from our own prior studies of IT security management. We categorized and combined these into a set of high level guidelines and identified the relationships between the guidelines and challenges in IT security management. We also illustrated the need for the guidelines, where possible, with quotes from additional interviews with five security practitioners. Our framework of guidelines can be used by those developing IT security tools, as well as by practitioners and managers evaluating tools.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; H.5.2 [**Information Interfaces and Presentation**]: UIs—*Interaction Styles*; H.5.3 [**Information Interfaces and Presentation**]: Group and Org. Interfaces—*Collaborative Computing*

General Terms

Human Factors, Security Management, Design

1. INTRODUCTION

IT security is an important issue for organizations that want to protect their information assets from threats inside or outside the organization. Previous studies show that, beside technological factors, human and organizational factors also impact IT security management(ITSM) [36, 9, 22]. Security practitioners (SPs) face challenges (discussed in detail in section 2.2) related to each of those factors [44]. In order to improve the effectiveness of IT security in an organization, these challenges must be addressed.

One way to address the challenges in ITSM is to develop effective technological solutions and tools to aid IT practitioners in managing security. A key factor that impacts the effectiveness of ITSM tools is their usability [12]. In this paper, we present a set of guidelines for ITSM tools based on the available literature and results of the HOT Admin project (see [22] for an overview of the project). Developing a set of guidelines specific to such tools is necessary, due to the importance of IT security in organizations and the evolving and competitive market of tools for managing it [8]. For each guideline, we identify the challenges that it can alleviate. We support the need for each guideline through literature and illustrate it through quotes from five SPs interviewed in the HOT Admin project. In addition, we propose a framework for classification of the guidelines. This framework can be used by tool developers to select appropriate guidelines when developing ITSM tools, as well as by SPs and their managers for evaluating such tools.

The rest of this paper is organized as follows. Next section presents background and related work. Section 3 describes the methodology we used to obtain and classify guidelines. Section 4 presents our framework of guidelines for ITSM tools, discussing each guideline in turn. Section 5 describes how to apply the guidelines and discusses limitations of our work and our plans for future research. Section 6 concludes.

2. BACKGROUND AND RELATED WORK

We provide background for our work from three different perspectives. First, we provide background on the development of design guidelines and how they can be used in practice. Second, we summarize those findings of the HOT Admin project, which this research is based on. Finally, we review literature related to guidelines for ITSM tools.

2.1 Generating Design Guidelines

User interface design entails a considerable investment by various stakeholders and design guidelines can help the stakeholders in the design process [39]. For example, from the guidelines, a system analyst can derive design requirements, a software designer can derive application-specific design rules, and a manager can make the interface design process more efficient. There are challenges and considerations when guidelines are applied and used [39]. First, it should be noted that not all guidelines are applicable to all tools. Therefore, developers should select a subset of guidelines that are applicable to the specific tool they are developing. Second, guidelines must be generally worded so that they might apply to many tools. Therefore, specific *design rules* should be derived from more general guidelines.

Smith and Mosier [39] suggest that every guideline development effort should begin and end by acknowledging the significant contributions of other people. Therefore, reviewing available literature on the subject under study is an essential part of developing guidelines. Also, guidelines can be based on experience—either practical, or derived through research [39]. In the available literature on guidelines, both approaches can be seen. For example, thorough literature searches have led to the development of guidelines for designing multi-media learning tools [19], and for designing systems to support co-located collaborative work on a tabletop display [38]. In contrast, Theng et al. [40] propose guidelines based on case studies of three digital libraries, while Baldonado and Woodruff [4], based on their experience, propose design guidelines for systems that use multiple views.

One large set of guidelines is the *Research-Based Web Design & Usability Guidelines* [30]. A survey of literature and other sources resulted in an initial set of guidelines for web usability. These were reviewed to eliminate duplicate, conflicting, and vague guidelines. Consequently, the relative importance and strength of evidence for each guideline was determined by external reviewers. The guidelines were enriched with graphical examples about how each guideline can be implemented. Finally, they were grouped and organized through a card sorting exercise with a group of web designers. These guidelines continue to be updated based on new literature.

The guidelines for ITSM tools that we present in this paper were derived through a combination of our own research results from the HOT Admin project and a thorough survey of related work. In our ongoing development of guidelines, we follow a process similar to that of [30]. This paper presents our initial set of guidelines; validation and enrichment of the guidelines by SPs and usability experts will follow. As suggested by Smith and Mosier [39], whenever possible, we provide examples of how each guideline can be realized.

2.2 The HOT Admin Project

The ongoing HOT Admin research project aims to investigate human, organizational, and technological factors of IT security from the perspective of SPs. The project goals are: (1) to devise a methodology for evaluating the usability of ITSM tools; and (2) to design effective technological solutions and guidelines to aid SPs [22]. The HOT Admin researchers have performed a participatory observation in one academic workplace and conducted interviews with 36 SPs. Data from the interviews and observation were analyzed according to several themes [17, 23, 9, 45]. The findings concerning challenges to ITSM [44] are important to our development of guidelines, as our goal is to address the challenges by proposing improvements for security tools.

One set of challenges in IT security arises from fairly ubiquitous human and cultural traits that become an issue, in particular when SPs need to interact with other stakeholders [44]. To begin with, a *lack of security culture* can challenge the modification of existing practices (e.g., multiple employees using the same account to access a system). *Lack of training* makes implementation of security controls difficult, as people are not well educated about best IT security practices. Further, *communication of security issues* can suffer from communication break-downs, usually because of different stakeholders having *different perceptions of risk.*

A second set of challenges in IT security are related to the characteristics of organizations [44]. Besides people having different perceptions of risk, establishing the organizational process of *risk estimation* is a challenge. The trade-off between security and business processes often results in *low priority of security*, which, combined with the costliness of IT security, leads to insufficient budgets. SPs are typically over-worked and *tight schedules* can lead to human errors or suboptimal security controls. Mergers, acquisitions of other organizations, and business partnerships all involve the challenge of *interaction with other organizations* that have different IT security needs, cultures, and practices. *Distribution of IT security* across the organization is commonplace; large organizations may have different IT departments, each of which is responsible for its own security. *Controlling access to data* is challenging as sensitive data is often distributed in organizations and accessed by many stakeholders. Finally, with an *open academic environment*, solutions need to allow for academic freedom in educational organizations.

A third group of challenges is related to technological issues [44]. The complex structure of computer networks (e.g., many nodes and users), and the need for different solutions (e.g., firewall, intrusion detection system, anti-virus) for managing IT security create a challenge of *technological complexity.* The frequent revelation of new *vulnerabilities* is a challenge, because SPs must deal with them or risk their security being compromised. The *mobility of access* to organizations' IT poses yet another security challenge.

2.3 Guidelines for IT Security Tools

Next, we briefly overview some of the main research efforts in developing design guidelines for ITSM tools. Sources in support of specific guidelines are given in Section 4.

As discussed above, the results of the HOT Admin project comprise one source for guidelines. While some of the research themes offer guidelines (e.g., [9, 17, 44, 45]), the provision of guidelines is not the main goal of these papers and they do not provide an integration of all guidelines.

Based on data collected from ethnographic field studies of system administrators, IBM researchers propose guidelines [20, 7, 6]. As there are many similarities between general IT practitioners and security practitioners [17], guidelines for general IT tools are often applicable to ITSM tools as well. Eser and Haber [26] also propose a small set of guidelines specifically for ITSM tools.

Chiasson et al. [12] combine results from usable security, ecological interface design, social navigation, and persuasive technology to propose an initial set of design principles for security management systems. How their principles might address the breadth of challenges has yet to be articulated.

3. METHODOLOGY

Our main research questions in this work are: (1) What are the characteristics of a good ITSM tool? (2) How can these characteristics be implemented in ITSM tools? (3) Which challenges in IT security are addressed by these characteristics? (4) How can we express these characteristics in the form of guidelines and organize them in a way that can be useful for developers? To answer these questions, we collected data from two different sources: related literature and the HOT Admin corpus of semi-structured interviews with SPs who use IT security tools.

To develop the core guidelines, we first selected a set of primary publications to analyze. This set contained publications from the HOT Admin project (4 papers) and publications about ITSM tools that we found important (14 papers, including those mentioned in Section 2.3). Using these sources, we started compiling the guidelines for ITSM tools. We included explicit guidelines as well as recommendations for improving security tools, good practices followed in a specific ITSM tool, and wish lists about tools. In this process we identified 164 guidelines.

After identifying the guidelines, we categorized them using a Grounded Theory [11]. First, we performed *open coding* using codes that emerged from the data itself [13]. Then we employed *axial coding* [13] to combine those open codes that are conceptually the same. As we had a large number of guidelines, we wanted to combine the guidelines in the way that would be more useful for tool developers. We therefore performed a card sorting exercise and grouped the guidelines according to the challenges that they address. This resulted in an early version of the framework, similar to that shown in Figure 1. To validate and refine the guidelines, we both broadened our survey and analyzed additional interviews.

To broaden the survey, we performed a more comprehensive literature search. We reviewed the papers published in well-known conferences related to the topic, performed keyword searches, and mined the references from our original set of 18 papers. The result of this search was a list of 56 papers. We then reviewed the papers and found another 22 papers that could contribute to our guidelines.

We analyzed five semi-structured interviews with SPs to find support for our guidelines and illustrative examples. The interviews are part of the HOT Admin corpus, but had not been analyzed when the HOT Admin papers cited in our survey were written. Participants included two security managers at a technology company (P30, P31), a security analyst at a telecommunications company (P32), a security consultant (P33), and a security analyst/manager at a second telecommunications company (P34). Each interview was 1-2 hours long, audio-recorded, and transcribed. In the interviews, SPs were asked about their tasks, their organizational model, the tools they used, and the ITSM-related challenges. It is worth mentioning that the interviews were not performed solely to gain knowledge about design guidelines for security tools; however, they did contain considerable information about ITSM tools. To analyze the interviews, we used the guidelines initially identified as codes (i.e., pre-defined codes constructed from prior materials [2]).

4. ITSM DESIGN GUIDELINES

We have developed a framework (Figure 1) for classifying the design guidelines for ITSM tools. Its main purpose is to aid developers in selecting the guidelines. Each layer of the framework addresses a different set of challenges. The lower layers contain the guidelines that are applicable to a larger set of tools, while the upper layers show guidelines that are more specific to a certain set of tools. For example, the lowest layer in the framework comprises general usability guidelines for ITSM tools. These guidelines are applicable to all ITSM tools, as well as other tools. The next two layers contain guidelines that are necessary due to the work environment of SPs, which is characterized by technological and organizational complexity. As most of the ITSM tools should work in complex technological environments, the guidelines in the technological complexity layer are applicable to most ITSM tools (but not security tools for end-users). The guidelines in the next layer deal with the organizational complexity of ITSM. These are subdivided into three groups: guidelines to address general communication challenges, guidelines applicable to tools used in a process that involves other stakeholders, and guidelines applicable to tools used by distributed SPs. The upper layer of the framework contains guidelines that are grouped based on task properties of the tool: guidelines for tools that require intensive configuration and deployment, and guidelines for tools used in a process that requires intensive analysis.

We next discuss the guidelines contained within each layer. For each guideline, we discuss the ITSM challenges addressed and cite the related work that supports its inclusion in our framework. When possible, we provide illustrative examples from participants and give alternatives of the guideline.

4.1 General Usability Guidelines

The first layer includes general usability guidelines and recommendations that are applicable to tools for SPs. When performing the card sorting exercise, we realized that many of the guidelines for security tools were based on general usability principles (e.g., [34, 39]). Because these guidelines were originally developed for more general tools and interfaces and are available in many different sources, we do not list all of them here. However, we give an example of a general usability guideline that is particularly important for ITSM tools: providing help and documentation to users.

In the literature, there are sets of guidelines about help and documentation features for of IT and ITSM tools. For example, tool documentation should be available on the Internet and searchable using search engines [20]. Several help features have been suggested for security tools [24]; although directed at tools for end-users, most of them are applicable to ITSM tools as well. These include providing context sensitive help, online help, wizards, light-weight help features, and social navigation. One technique, safe staging, may not be as useful if the tool will only be used by expert users.

4.2 Technological Complexity Guidelines

There are multiple challenges related to technical complexity, including *mobile access* and *vulnerabilities* [44]. We next present guidelines that can address these challenges.

4.2.1 *Make Tools Combinable*

SPs must often use multiple tools to perform a single task [9], but the process of combining tools to perform a task is not

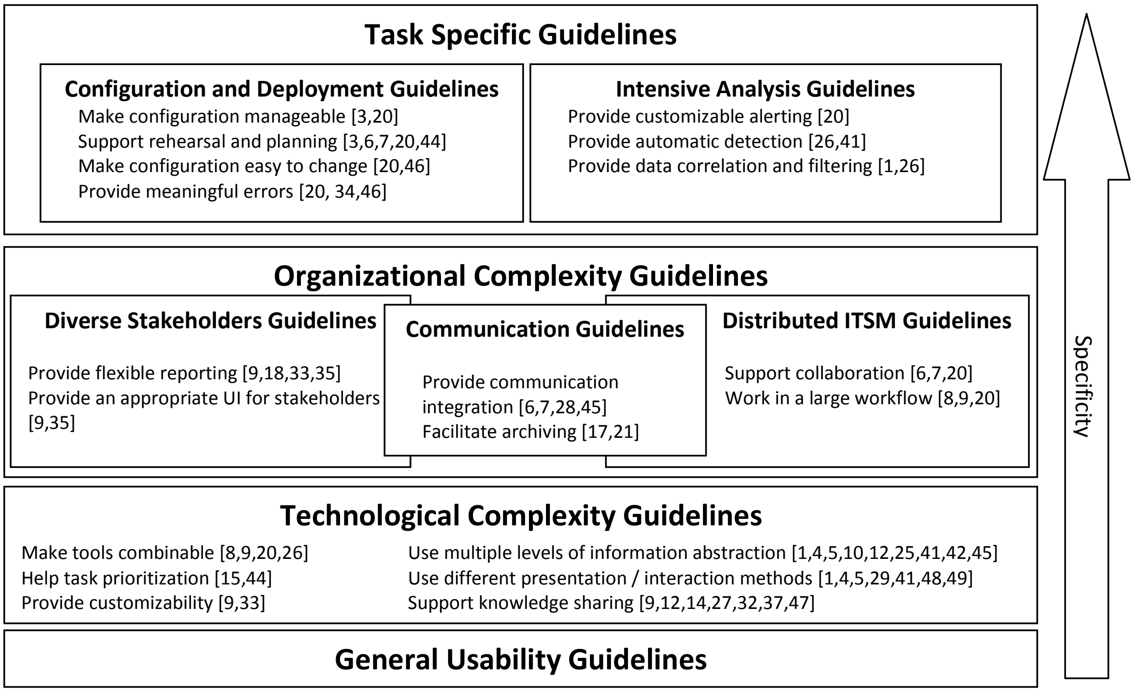
Task Specific Guidelines
Configuration and Deployment Guidelines
Make configuration manageable [3,20]
Support rehearsal and planning [3,6,7,20,44]
Make configuration easy to change [20,46]
Provide meaningful errors [20, 34,46]
Intensive Analysis Guidelines
Provide customizable alerting [20]
Provide automatic detection [26,41]
Provide data correlation and filtering [1,26]
Organizational Complexity Guidelines
Diverse Stakeholders Guidelines
Provide flexible reporting [9,18,33,35]
Provide an appropriate UI for stakeholders [9,35]
Communication Guidelines
Provide communication integration [6,7,28,45]
Facilitate archiving [17,21]
Distributed ITSM Guidelines
Support collaboration [6,7,20]
Work in a large workflow [8,9,20]
Technological Complexity Guidelines
Make tools combinable [8,9,20,26]
Help task prioritization [15,44]
Provide customizability [9,33]
Use multiple levels of information abstraction [1,4,5,10,12,25,41,42,45]
Use different presentation / interaction methods [1,4,5,29,41,48,49]
Support knowledge sharing [9,12,14,27,32,37,47]
General Usability Guidelines
Specificity

Figure 1: Framework of design guidelines for IT security management tools

well supported by available tools [8, 26]. As one of our participants illustrated: "So the vendors themselves are looking at things in isolation instead of looking at it as a whole thing that needs to be addressed" (P33). Another described challenges when using multiple tools: "We are really, really having a problem at correlating output from all these tools. At the beginning they were using three or four, it was easy to manually correlate, but when they started hitting six, seven, eight, plus, it was very difficult to correlate because the outputs are all different" (P34). This participant also mentioned that development of a console to configure and execute 17 vulnerability analysis tools resulted a significant decrease in the time needed to perform an analysis task (from 10-15 days, to two days).

Combining tools in an *ad hoc* fashion is a kind of *bricolage*; it is recommended that tools should survive in an arena of bricolage [9]. Vendors should standardize event formats to permit integration of tools [26]. Standardized configuration and logging formats will allow files from different tools to be searched and correlated together [20]. Another option is to provide APIs/plug-ins to facilitate integration of tools into system-wide monitoring or management meta-tools [20].

4.2.2 Support knowledge sharing

As SPs perform their tasks within complex technological environments, a great deal of knowledge is created. This knowledge is either kept in the mind of the security practitioner, or written in notes or documents, or kept in the form of executable scripts [9]. This knowledge is a valuable asset and can be used in the future by the same or other SPs; it therefore should be kept and managed [27]. This knowledge can be managed at two levels: among SPs in the organization or among all the users. Therefore, security tools should facilitate knowledge management at different levels. To support knowledge sharing, SPs can use databases, Microsoft SharePoint sites, document management systems, or Wikis [37, 47, 32]. This practice is illustrated by one participant who mentioned: "We have an IT manual which is kept up to date electronically and hard copy. We have our SharePoint site where they can go and everything is at their fingertips. It links them to every single place they need to know how to go to" (P33). Another form of sharing is social navigation [12, 14] which is mainly practiced over the Internet. Although arising from technological complexity, the need to support knowledge sharing is closely related to organizational complexity as described in Section 4.3.

4.2.3 Use different presentation/interaction methods

Presenting information in multiple views or presentation formats can facilitate investigation of a single conceptual entity [4]. Textual and speech data is sequentially processed through auditory cognitive functions, while graphical data has the advantage of using parallel visual and spatial functions [5]. Therefore, graphical data results in faster situational awareness and effective identification of patterns and vulnerabilities in network. Furthermore, using different presentations of the same data can help situation awareness through a reduction in the high cognitive load characteristic of ITSM [48]. In the related literature, this guideline often accompanies a proposal for different visualization methods for a large set of data. For example, one proposed visualization method for intrusion detection system (IDS) alarms is to show alarms in a two-dimensional space (y-axis: IP address, x-axis: time) [1]. Three-dimensional space has also been proposed for IDS data (color, opacity, shape) [29]. Different levels of detail can allow SPs to zoom in for more details [29]. Visualization of the network data can reduce the time and training required for network traffic analysis [5]. The combination of concurrent textual and visual interfaces has been advocated for security tools [49] and IDSs as each interface has its own strengths and weaknesses [41]. This guideline may be particularly important for those tools that involve intensive analysis, as described in section 4.4.2.

4.2.4 *Use multiple levels of information abstraction*

Vicente and Rasmussen [42] suggest using an abstraction hierarchy in order to support operators of complex systems during unanticipated events. As SPs need to deal with vulnerabilities and unanticipated scenarios in complex network environments, tools can follow Vicente's method of showing the system at different levels of abstractions, with the Ecological Interface Design (EID) framework. Other researchers suggest using EID in the design of ITSM tools in general [12] and for network monitoring tools in particular [10]. Presenting information at different levels of abstraction to different stakeholders can help prevent disclosure of confidential information by presenting it to each stakeholder at an appropriate level [45]. It can also prevent miscommunication by providing information appropriate to the stakeholder's level of security knowledge [45].

In many tools, presenting information at different levels of abstraction is realized by providing an interface with two views: overview and details. A study about reading electrical documents [25] suggests that presenting information at different levels of detail can reduce user errors. Similarly, using multiple levels of abstraction, as well as visualization, have been found to reduce the time and training required for network traffic analysis [5].

When there is diversity in levels of abstraction, different presentation formats (Section 4.2.3) can be used [4]. For example, both textual and graphical interfaces for IDSs have been used to present information at different levels of abstraction [41]. Similarly, a visualization technique for IDS provides an overview of its alarms with details on demand [1].

4.2.5 *Provide Customizability*

As SPs frequently deal with unpredictable situations (e.g., new vulnerabilities), an essential feature for ITSM tools is to be customizable [9]. This need is illustrated by one of our participants (P30): "For the reasons why I have built some stuff from hand, we'll say that no, they don't do everything that I need them to do. So sometimes I do need to custom craft something or I need to automate something. Or I need to do something maybe that the tool doesn't do." In a comparison of security analysis tools for SIP-based VoIP systems, one important criterion was the ability to define new and customized test-cases [33].

4.2.6 *Help Task Prioritization*

SPs frequently must deal with many competing priorities. A recent survey [15] found that one of the main factors that frustrates SPs is wasting time and that they need better planning and organization. Therefore, it is important for tools to facilitate the process of planning and prioritization. Planning facilities can be implemented in different ways. At the most basic, a tool could afford note-keeping functionality so that SPs can write down their priorities with regards to the tool. With some intelligence, a tool could help to prioritize vulnerabilities based on their criticality [44].

4.3 Organizational Complexity Guidelines

Several aspects of organizational complexity must be addressed by the guidelines. SPs need to communicate with many stakeholders, including both other SPs and diverse stakeholders within the organization [45]. We first present guidelines that address general communication challenges and then present the guidelines that specifically address the challenge of dealing with diverse stakeholders. Finally, we present guidelines that address the challenges arising from the distribution of security tasks across multiple SPs.

4.3.1 *Communication Guidelines*

Ineffective communication is a contributing factor to human errors [31]. SPs need to communicate with other stakeholders during many activities, and the current tools do not provide sufficient communication support for SPs [45]. Additionally, it is necessary for SPs to communicate both with tools and with each other [9, 44, 37]. Tools that facilitate communication can address challenges of *communication of security issues*, *distribution of IT management*, *interaction with other organizations*, and *different perceptions of risk*.

4.3.1.1 *Provide Communication Integration*

One way for tools to facilitate communication is to allow for integration with communication media. Tools should have communication facilities to allow collaboration between different users [7, 6]. Tools can reduce communication overhead between different stakeholders by showing relevant security configuration information to different stakeholders [45]. One important feature, whether communication is between a tool and a user or between users, is to support a secure method of communication [28].

The need to support communication between tools and users is illustrated by one of our participants when discussing a network monitoring tool (P33): "You hate to find out you have a problem when you are actually working; rather get paged at night and be able to fix it before they show up." Tools should be integrated with different channels (e.g., email, text messages, web site) [45]. Mobile communication modalities (e.g., pagers, Blackberry email) should be integrated into the solutions [6]. Furthermore, tools should be configurable as to the destination and stakeholder to which these messages (e.g., alarms, logs) should be sent [45].

4.3.1.2 *Facilitate Archiving*

Tools should facilitate keeping track of communication and information related to tools. Practicing this guideline has two benefits. First, keeping a record of communication between different stakeholders is already practiced by SPs; this may be due to the need for SPs to adhere to legislation [17]. It is also illustrated by one participant (P33): "So we have archives with backup tapes—we have the Cadillac of backup tapes for our kind of organization because we have a thing you can walk into practically—so we keep everything." If tools provide support for this need, they can help remove the burden of archiving and managing communication. A second benefit of archiving is to keep the information and knowledge that is generated during one project or incident [21]. This information can be used in future incidents to analyze the trends in network, or it can be used as a knowledge base (previously discussed in Section 4.2.2).

4.3.2 Diverse Stakeholders Guidelines

One important organizational challenge of IT security is the involvement of various stakeholders within the organization [44], and some aspects of security management may involve non-experts [9]. Furthermore, effective communication of security issues to different stakeholders is an important factor to be considered. "Not realizing the core importance of information security awareness amongst users" is considered one of the "deadly sins" of ITSM [43]. We next present guidelines that are mainly aimed at addressing communication challenges (*communication of security issues*, *different perceptions of risk*) that result from diverse stakeholders.

4.3.2.1 Provide an Appropriate UI for Stakeholders

ITSM tools often have many types of end-users including experts such as SPs and less technical administrators and managers [9]. Each category of users may have its own preferences and needs in terms of the user interfaces. As illustrated by one of our participants (P34): "We actually use the command line interface route and we try to keep it as simple as possible because we were putting another layer on top of it we couldn't go into the graphical one. But sometimes clients want graphical stuff. Especially if they are not 100% techie, it's easier." Therefore tools should provide appropriate user interfaces based on the user's expertise and needs. One suggestion, when developing a UI for ITSM tools that will be used by managers, is to provide an overview early with as little information as possible and provide further details on demand [35]. This guideline relates to Sections 4.2.3 and 4.2.4, as different stakeholders require different presentations of data or different levels of detail in their user interface.

4.3.2.2 Provide Flexible Reporting

One aspect of flexible reporting is generating reports that are customized to contain information for a specific stakeholder. For example, reports that are aimed at managers should be concise and mainly focus on business objectives and the effectiveness of the organization in reaching them [18, 35]. This is illustrated by one of our participants (P34): "And the CEO is comfortable talking to me because I am talking his language. I am talking your return on investment, . . . and those are the terms that I use quite often. I do it deliberately. It's a technique that I've learned and I've used shamelessly." Also, the report may be packaged differently depending on the type of stakeholder. For example, one participant talked about packaging a report for managers (P33): "It's got to have color and it's got to be flashy so that they'll pay attention. They don't want to read a ten page document on anything. They want a quick learn." From a different angle, two of our participants (P34, P30) discussed that reports should contain constructive recommendations that are simply represented (e.g., in a table). In addition, reports can be packaged based on predefined templates or standard frameworks like Sarbanes Oxley bill or IS 17799 [18]. Garigue and Stefaniu [18] classify reports into four categories and provide examples of the important reports that can be generated in each category.

Another aspect of the flexible reporting is making reports accessible by different stakeholders. To realize this, reports should be easily distributable and accessible across the organization. For example, reports can be generated in standard formats like HTML, PDF, and spreadsheets [9]; generating reports in the web format is considered an important feature for security analysis tools [33].

The flexible reporting guideline relates to Section 4.2.6 because a well sorted report can help prioritization. Also the guideline relates to Section 4.2.3, as providing different presentations of data will make reports more understandable. Finally, flexible reporting relates to Section 4.2.4, as the reports generated by a tool should provide an appropriate level of detail based on the intended audience.

4.3.3 Distributed ITSM Guidelines

The guidelines presented next address the challenge of *distribution of IT management* [44]. In many organizations ITSM is distributed across multiple SPs [9], either informally or through an official distributed organizational model for ITSM [23]. In these organizations, SPs need to collaborate with each other, as well as other stakeholders, to perform tasks [45]. Tools should function as part of a larger workflow and provide support for collaborating and sharing.

4.3.3.1 Work in a Large Workflow

One of the important needs of SPs while working under distributed ITSM is to be able to automatically distribute tasks. Security tools should follow the way corporate networks have evolved and become integrated together [8]. To allow collaboration among stakeholders, there is a need for workflow support for the varying roles of different individuals [9]. One of our participants (P32) desired an access control platform that supported the workflow of granting access to a user: from the end user request, to the person in charge of authorization, to the administrator making changes to the security controls. Shifts in responsibilities could be encoded in scripts with a new sysadmin automatically notified when it is their turn and the pertinent interface displayed [20].

4.3.3.2 Support Collaboration

One important feature of tools that can help collaboration is to provide a shared view of the system state [7, 6, 20]. Tools should formally support sharing by showing which users are currently working with system and what they are doing [20]. In addition, sharing can be supported with proper approval and authentication [7, 6]. Another important aspect of collaboration is to provide support for grounding new participants as quickly as possible when they join the activity [20]. This guideline is an extension of the knowledge sharing guideline (Section 4.2.2).

4.4 Task Specific Guidelines

This layer contains guidelines that may or may not be applicable, depending on the nature of the application. The first set of guidelines is specific to applications that require intensive configuration, particularly during deployment. The second set is specific to applications that require SPs to perform intensive analysis.

4.4.1 Configuration and Deployment Guidelines

SPs must often perform complex configuration of tools, particularly during deployment. Due to the *technological complexity* of ITSM and its tools, the task of configuration can require a great deal of effort [44]. A second challenge that impacts configuration and deployment is *vulnerabilities* [44]. To deal with frequent vulnerabilities, SPs need to patch systems often; however, patching a network of thousands of nodes is tedious work that can be very costly. For example, manually deploying a patch on a 1000-node network can cost as much as $1M [3]. Because security has a *low priority* within many organizations [44], SPs may be urged to complete configuration and deployment as quickly as possible, and without compromising availability and performance. We next present several guidelines aimed at dealing with these challenges.

4.4.1.1 Make Configuration Manageable

As described above, SPs frequently need to apply configuration changes to hundreds of nodes in a network or deploy nodes at a similar scale. This complex process should be done very quickly and accurately. Therefore, tools should enable SPs to automate and manage this process, as well as control its details. To realize this, tools should provide progress indicators, forecast the deployment process, perform operations in an asynchronous non-blocking manner, and provide history and detailed steps of the executed operation [20]. Tools should also support change roll-back [3].

4.4.1.2 Support Rehearsal and Planning

As SPs work with complex and critical systems, changes in configuration may have unanticipated outcomes that cannot be tolerated by other stakeholders. For example, security patches are not usually tested for all environments [3]. Also, Werlinger et al. [44] illustrate how a security patch that decreased the performance of an application triggered conflict between SPs and internal users. Therefore, SPs should be very careful in deploying new solutions, changing configuration, or applying patches. System administrators practice rehearsal and planning to avoid unanticipated events on production systems [6, 20]. They first rehearse the operation on a test system and then apply it to the production system. Security tools that require extensive configuration should support rehearsal and planning practices. It should be easy to build a test system with various degrees of fidelity to the production system, and it should be easy to validate the results of the test system [6].

Also, tools should support migration of scripts/operations from test to production environments [20]. Logging each step of the procedure and providing facilities to compare the outputs from test and production environments would facilitate rehearsal and planning [7]. Virtual environments (e.g., using tools like VMware) can assist with testing [3].

The rehearsal and planning guideline is related to making the configuration process manageable (Section 4.4.1.1). The rehearsal process can be completed more easily and with less overhead if tools support features like undo. Furthermore, providing forecasting of the deployment can be a useful indicator for comparing the rehearsal with actual execution.

4.4.1.3 Make Configuration Easy to Change

One of the tasks of SPs is to change the configuration of the system. Configuration frequently requires dealing with many parameters, some of which are unknown to the security practitioner. Therefore, tools should provide facilities that help SPs change configuration of the system easily. To realize this, tools should provide commented configuration files and/or group related parameters together in high-level profiles [20], so that a change in the profile would change all related parameters automatically. Also, tools should provide a quick tuning option that allows batch configuration of parameters [46].

4.4.1.4 Provide Meaningful Errors

Although providing meaningful errors is a standard usability practice [34], we re-iterate the guideline here as configuration and deployment of complex systems is particularly error prone. To ease the process, particular care should be taken for any error messages generated by tools during configuration and deployment so that it is presented to the user in a meaningful way. For example, insufficiently meaningful error messages caused delays during installation of an IDS in one academic organization [46]. One suggestion is that tools should provide help in case of errors or alerts, instead of presenting cryptic messages [20].

4.4.2 Intensive Analysis Guidelines

Investigation of attacks and vulnerabilities is one of the most important and challenging tasks for SPs [44] and requires periods of intensive analysis. To deal with the challenge of *vulnerabilities*, tools should support SPs in the investigation tasks. As SPs must conduct analysis within the constraints of *tight schedules* [44], tools should provide mechanisms to reduce the number of false positives (FPs) because FPs have to be investigated. The next three guidelines are applicable to tools which require SPs to perform intensive analysis.

4.4.2.1 Provide Customizable Alerting

Many security tools that monitor systems, generate alarms communicated to SPs. Tools should provide customizable thresholds for generating alarms and selectable destinations for sending alarms (e.g. pager, email, console) [20]. SPs should be able to suppress alarms with lower priority [20]. As mentioned by one of our participants (P32): “Given that I had knowledge of the perimeter security systems that were protecting these systems internally in the organization, I modified that critical level [of the tool].”

4.4.2.2 Provide Automatic Detection

SPs need to find attacks or unusual behavior patterns in large amount of logs and data that are linked together [26]. To help SPs perform their tasks more effectively, tools can provide automation in detecting problems [26]. Application of data mining and other analytic methods in activity classification, analysis, and noise reduction can help; automatic detection could be implemented as software agents or *bots* that handle obvious cases and notify SPs about critical ones [26]. Tools can use intelligent pattern recognition techniques to find salient patterns [41].

4.4.2.3 Provide Data Correlation and Filtering

During analysis, SPs frequently need to collect data from several sources and then correlate it [26]. For example, correlating alarms of different IDSs can reduce the number of false alarms [1]. As discussed by one participant (P32): "These tools generate general or global reports based on what they are analyzing, right? And, I took those reports, with other tools I complemented like NMAP to do the same analysis, and I was checking and corroborating that they effectively correspond." Security tools can improve the process of data correlation by providing required filtering to reduce the large quantity of data, providing output in standard formats that can be shared between different tools, providing facilities to deal with the problem of out-of-sync clocks in correlating time-stamped data from different sources, and providing facilities to automate the process of data correlation [26].

5. DISCUSSION

5.1 Applying the Guidelines

These guidelines can be used for multiple purposes. They can be used by developers as they compile requirements for applications with SPs as end-users. Consideration of the guidelines may also be helpful when developing rich use case scenarios to ensure that the scenarios address the technological and organizational challenges of the intended operational environment. The guidelines could also be used by SPs and managers as they evaluate tools in the context of the challenges inherent within their organization. In this case, the guidelines should be treated as high-level criteria. If a criterion is not met in some way, then the application may have room for improvement.

Whether designing an application, deciding which one to acquire, or evaluating an application, the sets of guidelines that are relevant to the application and to the situation should be considered. The guidelines are grouped by whether they are (1) generally applicable, (2) relevant to technological or organizational characteristics of the operational environment, or (3) relevant to task-specific challenges. *General usability guidelines* apply to all ITSM-relevant applications. We argue that the guidelines to address *technological complexity* are also applicable to all ITSM-relevant applications. Not only is the technological environment of ITSM changing with the advent of new technologies, but the rate of change in ITSM is faster than in general IT [17]. That is, technological complexity is characteristic of ITSM. As discussed next, the applicability of the remaining guidelines will depend on the organizational environment and the specifics of the tasks the application will perform. Not all the guidelines will be applicable in all situations.

The organizational environment in which the application is deployed may be such that the guidelines to help *organizational complexity* are relevant. An organization may be large with many cooperating SPs, or small with an IT department consisting of a "one-man shop." Depending on the intended use of the tool in that environment, guidelines to help *multiple stakeholders*, or guidelines to help *distributed ITSM*, or both may apply. A developer of a tool that will be used in a "one-man shop" may not need to consider guidelines that address organizational complexity, particularly those guidelines that address *distributed ITSM*. However, a tool—such as an IDS—installed in a large organization would likely require ongoing distributed cooperation between SPs [46] and therefore benefit from the guidelines on *distributed ITSM*. In contrast, if an access control system were to be implemented in the same large organization, it would require a great deal of initial cooperative consultation to establish job roles and their corresponding privileges. For such an application, the organizational complexity would likely focus more on multiple stakeholders than on distributed ITSM, so it would benefit from guidelines to help *multiple stakeholders*.

Task specific challenges may be addressed by guidelines to help *intensive analysis* or guidelines to help *configuration and deployment*. For example, an IDS is typically difficult to configure and also requires intensive ongoing analysis [46], while a network scanning application (e.g., Nessus) requires intensive analysis, but typically does not require much configuration. The network scanning application would therefore only need the guidelines to help *configuration and deployment* while the IDS could benefit from consideration of all the guidelines in this layer.

5.2 Limitations and Future Work

While we have considered 77 publications and integrated 39 of them into our survey, the framework is still under development. As research continues, more guidelines may come to light. We will maintain a website (http://www.hotadmin.org/guidelines.html), which we will update as our understanding of the guidelines matures.

We have attempted to show the relationship between guidelines and challenges to ITSM. These relationships could help SPs and tool developers to decide about the importance of each guideline. However, the understanding of each guideline importance should be investigated further. Based on Koyani et al.'s [30] successful experience with the development of design guidelines, we plan to survey SPs on the importance of each guideline.

We have also identified the methodology used in each cited source generating guidelines, as well as whether the guidelines were generated by studying specific user populations (e.g., security practitioners, system administrators) and whether they were generated considering specific security tools (e.g., IDS). Our next iteration of the guidelines will include an analysis of the strength of evidence for the guidelines, which will help SPs and tool developers evaluate the validity and generalizability of the guidelines.

The guidelines in our framework are high level and blur the boundaries between usability, organizational usability [16], and utility. Although our ongoing work and that of other researchers will doubtless refine the framework over time, our framework provides a schema with which to focus research attention. Beyond improving the breadth and importance of the guidelines, it would be interesting to study how such guidelines are applied in practice. Each guideline warrants deeper study into how it is already practiced, and how it could be practiced. To that end, we plan to conduct case studies of organizations using the guidelines during tool development and evaluation.

6. CONCLUSION

In this paper, we provided the result of our preliminary survey on design guidelines for ITSM tools. The primary sources for the guidelines are recommendations about ITSM tools available in the literature; we have augmented these from our own experiences interviewing SPs in the HOT Admin project. We have gathered the different recommendations and combined them into a framework of high-level design guidelines for ITSM tools. This framework can be used by tool developers, as well as by SPs and managers evaluating security tools. To justify the guidelines, we provided empirical evidence of their need. In addition, we identified relationships between the guidelines and known challenges in ITSM. These relationships can help users of the framework determine the importance of each guideline for their tools. We have identified several areas of future work to help refine the guidelines, including determining their relative importance, validity, and generalizabiity.

7. ACKNOWLEDGMENTS

This work would not be possible without access to the HOT Admin study participants. Sonia Chiasson, Alain Forget, Julie Thorpe, and Robert Biddle of Carleton helped us to review and brainstorm an early version of the guidelines. HOT Admin project members provided helpful comments on preliminary versions of this paper. Comments from anonymous reviewers helped us to improve the paper further. The HOT Admin project is supported by the NSERC Strategic Partnership Program, grant STPGP 322192-05.

8. REFERENCES

[1] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko. IDS RainStorm: Visualizing ids alarms. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, pages 1–10, Minneapolis, MN, USA, 2005. IEEE Computer Society.

[2] P. A. A. Amanda Jane Coffey. *Making Sense of Qualitative Data: Complementary Research Strategies.* SAGE Publications, 1996.

[3] C. Andrew. The five ps of patch management: Is there a simple way for businesses to develop and deploy an advanced security patch management strategy? *Computers & Security*, 24(5):362–363, 8 2005.

[4] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 110–119, Palermo, Italy, 2000. ACM.

[5] R. Ball, G. A. Fink, and C. North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, Fairfax, VA, USA, 2004. ACM.

[6] R. Barrett, P. P. Maglio, E. Kandogan, and J. Bailey. Usable autonomic computing systems: The system administratorsṕerspective. *Advanced Engineering Informatics*, 19(3):213–221, 2005.

[7] R. Barrett, M. Prabaker, and L. Takayama. Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. In *CSCW '04*, pages 388–395, Chicago, IL, USA, 2004.

[8] B. Beal. IT security: the product vendor landscape. *Network Security*, 2005(5):9–10, 5 2005.

[9] D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels, and B. Fisher. Towards understanding IT security professionals and their tools. In *SOUPS '07: Proceedings of the 2007 Symposium On Usable Privacy and Security*, pages 100–111, Pittsburgh, Pennsylvania, July 18-20 2007. ACM.

[10] C. M. Burns, J. Kuo, and S. Ng. Ecological interface design: a new approach for visualizing network management. *Comput. Netw.*, 43(3):369–388, 2003.

[11] K. Charmaz. *Constructing Grounded Theory.* SAGE publications, 2006.

[12] S. Chiasson, P. C. van Oorschot, and R. Biddle. Even experts deserve usable security: Design guidelines for security management systems. In *SOUPS Workshop on Usable IT Security Management (USM)*, Pittsburgh, PA, July 2007.

[13] J. W. Creswell. *Qualitative Inquiry and Research Design : Choosing among Five Traditions.* SAGE Publications, July 1997.

[14] P. DiGioia and P. Dourish. Social navigation as a model for usable security. In *SOUPS '05: Proceedings of the 2005 Symposium On Usable Privacy and Security*, pages 101–108, Pittsburgh, Pennsylvania, 2005. ACM.

[15] B. Dijker. A day in the life of system administrators. http://sageweb.sage.org, June 2006.

[16] M. Elliott and R. Kling. Organizational usability of digital libraries: Case study of legal research in civil and criminal courts. *American Society for Information Science*, 4(11):1023–1035, 1997.

[17] A. Gagné, K. Muldner, and K. Beznosov. Identifying differences between security and other IT professionals: a qualitative analysis. In *HAISA'08: Human Aspects of Information Security and Assurance*, pages 69–80, Plymouth, England, July 8-9 2008.

[18] R. Garigue and M. Stefaniu. Information security governance reporting. *EDPACS*, 31(6):11–17, 2003.

[19] T. Grunwald and C. Corsbie-Massay. Guidelines for cognitively efficient multimedia learning tools: educational strategies, cognitive load, and interface design. *Academic medicine*, 83(3):213–223, 2006.

[20] E. M. Haber and J. Bailey. Design Guidelines for System Administration: Tools Developed through Ethnographic Field Studies. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer Human Interaction for the Management of Information Technology*, pages 1–9. ACM, 2007.

[21] C. A. Halverson. The value of persistence: A study of the creation, ordering and use of conversation archives by a knowledge worker. In *HICSS '04: Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 1–10, Washington, DC, USA, 2004. IEEE Computer Society.

[22] K. Hawkey, D. Botta, R. Werlinger, K. Muldner, A. Gagne, and K. Beznosov. Human, Organizational, and Technological Factors of IT Security. In *CHI'08 extended abstract on Human factors in computing systems*, pages 3639–3644, Florence, Italy, 2008.

[23] K. Hawkey, K. Muldner, and K. Beznosov. Searching

for the Right Fit: Balancing IT Security Model Trade-offs. *Special Issue on Useful Computer Security, IEEE Internet Computing*, 12(3):22–30, 2008.

[24] A. Herzog and N. Shahmehri. User help techniques for usable security. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer Human Interaction for the Management of Information Technology*, pages 93–102, Cambridge, Massachusetts, 2007. ACM.

[25] K. Hornbaek and E. Frokjaer. Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 293–300, New York, NY, USA, 2001. ACM.

[26] E. Kandogan and E. M. Haber. Security administration tools and practices. In L. F. Cranor and S. Garfinkel, editors, *Security and Usability: Designing Secure Systems that People Can Use*, chapter 18, pages 357–378. O'Reilly Media, Inc., 2005.

[27] S. Kesh and P. Ratnasingam. A knowledge architecture for it security. *Commun. ACM*, 50(7):103–108, 2007.

[28] G. Killcrece, K.-P. Kossakowski, R. Ruefle, and M. Zajicek. Organizational models for computer security incident response teams (CSIRTS). Technical Report CMU/SEI-2003-HB-001, 2003.

[29] A. Komlod, P. Rheingans, U. Ayachit, J. Goodall, and A. Joshi. A user-centered look at glyph-based security visualization. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, pages 21–28, Minneapolis, MN, USA, 2005.

[30] S. J. Koyani, R. W. Bailey, and J. R. Nall. *Research-Based Web Design & Usability Guidelines.* U.S. Dept. of Health and Human Services, 2006.

[31] S. Kraemer and P. Carayon. Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists. *Applied Ergonomics*, 38:143–154, 2007.

[32] C. P. Lee and J. A. Copeland. Flowtag: a collaborative attack-analysis, reporting, and sharing tool for security researchers. In *VizSEC '06: Proceedings of the 3rd international workshop on Visualization for computer security*, pages 103–108, Alexandria, VA, USA, 2006. ACM.

[33] S. McGann and D. C. Sicker. An analysis of security threats and tools in SIP-based VoIP systems. In *2nd VoIP Security Workshop*, pages 1–8, Washington DC, USA, June 2005.

[34] J. Nielsen. Applying discount usability engineering. *IEEE Software*, 12(1):98–100, 1995.

[35] M. Nohlberg and J. Backstrom. User-centred security applied to the development of a management information system. *Information Management & Computer Security*, 15(5):372–381, 2007.

[36] R. H. Rayford B. Vaughn Jr. and K. Fox. An empirical study of industrial security-engineering practices. *The Journal of Systems and Software*, 61:225–232, 2001.

[37] Y. Rogers. Ghosts in the network: distributed troubleshooting in a shared working environment. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 346–355, Toronto, ON, Canada, 1992. ACM.

[38] S. D. Scott, K. D. Grant, and R. L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.

[39] S. L. Smith and J. N. Mosier. Guidelines for designing user interface software. Technical Report ESD-TR-86-278, The MITRE Corporation Bedford MA, August 1986.

[40] Y. L. Theng, E. Duncker, N. Mohd-Nasir, G. Buchanan, and H. W. Thimbleby. Design guidelines and user-centred digital libraries. In *ECDL '99: Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, pages 167–183, London, UK, 1999. Springer-Verlag.

[41] R. S. Thompson, E. M. Rantanen, W. Yurcik, and B. P. Bailey. Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1205–1214, San Jose, California, USA, 2007. ACM.

[42] K. Vicente and J. Rasmussen. Ecological interface design: theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, Jul/Aug 1992.

[43] B. von Solms and R. von Solms. The 10 deadly sins of information security management. *Computers security*, 23(5):371, 2004.

[44] R. Werlinger, K. Hawkey, and K. Beznosov. Human, Organizational and Technological Challenges of Implementing IT Security in Organizations. In *HAISA'08: Human Aspects of Information Security and Assurance*, pages 35–48, Plymouth, England, July 8-9 2008.

[45] R. Werlinger, K. Hawkey, and K. Beznosov. Security practitioners in context: their activities and interactions. In *CHI '08 extended abstracts on Human factors in computing systems*, pages 3789–3794, Florence, Italy, 2008.

[46] R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, and K. Beznosov. The challenges of using an intrusion detection system: Is it worth the effort? In *SOUPS '08: Proceedings of the 2008 Symposium On Usable Privacy and Security*, pages 107–116, Pittsburgh, Pennsylvania, July 23-25 2008.

[47] K. F. White and W. G. Lutters. Midweight collaborative remembering: wikis in the workplace. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer Human Interaction for the Management of Information Technology*, pages 111–112, Cambridge, MA, USA, 2007. ACM.

[48] W. Yurcik, J. Barlow, and J. Rosendale. Maintaining perspective on who is the enemy in the security systems administration of computer networks. In *ACM CHI Workshop on System Administrators Are Users, Too. Proceedings of the Tenth Americas Conference on Information Systems*, 2003.

[49] W. Yurcik, R. S. Thompson, M. B. Twidale, and E. M. Rantanen. If you can't beat 'em, join 'em: combining text and visual interfaces for security-system administration. *Interactions*, 14(1):12–14, 2007.